



INSTITUTO/S: Instituto de Tecnología e Ingeniería

CARRERA/S: Tecnicatura Universitaria en Informática / Licenciatura en Informática

MATERIA: Estructuras de datos

NOMBRE DEL RESPONSABLE DE LA ASIGNATURA: Dr. Sergio Alberto Gonzalez

EQUIPO DOCENTE: Dr. Sergio Gonzalez, Esp. Martin Pustilnik

CUATRIMESTRE: 2do.

AÑO: 1ero.

PROGRAMA N°: Tecnicatura: 5 / Licenciatura: 5

FECHA (de aprobación): 15/02/2022

Instituto/s: Instituto de Tecnología e Ingeniería

Carrera/s: Tecnicatura Universitaria en Informática. Licenciatura en Informática

Nombre de la materia: Estructuras de datos

Responsable de la asignatura y equipo docente: Dr. Gonzalez Sergio, Esp. Martin Pustilnik

Cuatrimestre y año: 2do del 1er año

Carga horaria semanal: 8 hs

Programa N°: Tecnicatura: 5 / Licenciatura: 5

Código de la materia en SIU: 752

Estructuras de datos

1. Fundamentación

El perfil del egresado/a de la carrera, apunta a un profesional de la construcción de software con parámetros de calidad, tanto en etapas de diseño como de programación, con una base conceptual sólida que permita adaptarse a proyectos de distinta índole y favorecer el trabajo en equipo y colaborativo. En este contexto, la materia toma un importante rol, ya que, en esta se adquieren conceptos centrales que luego se utilizarán a lo largo de la carrera y en la práctica profesional.

2. Propósitos y/u objetivos

Se desea que el/la estudiante:

- Comprenda los conceptos de programación estructurada y modular.
- Comprenda el concepto de dato, tipos de datos primitivos y estructuras de datos, y su importancia e interrelación con la resolución algorítmica de un problema.
- Conozca la interfaz de una estructura de datos y su utilización para la resolución de problemas.
- Se familiarice con los conceptos de ámbito y de pasaje de parámetros por valor o por referencia.
- Pueda resolver problemas mediante programas recursivos, y entienda la diferencia entre una solución recursiva y una iterativa.
- Comprenda el concepto de asignación dinámica de memoria y pueda escribir programas que realicen un manejo dinámico explícito de memoria en forma adecuada.
- Conozca las estructuras de datos básicas y su interfaz: Vectores, matrices, pilas, colas, listas, árboles, diccionarios, hashing, grafos, etc.
- Comprenda el funcionamiento interno de estructuras de datos básicas y maneje principios básicos del diseño de la interfaz.
- Entienda la noción de implementación de una estructura de datos, y de su eficiencia, y sea capaz de implementar las interfaces vistas anteriormente con alternativas variadas en eficiencia.

- Se familiarice con las tareas de compilar y vincular programas para lograr un ejecutable.
- Incorpore la elección de la estructura de datos como una parte de la solución de un problema, en relación con la resolución algorítmica y la programación en computadora.
- Utilice e internalice buenas prácticas de programación, en relación a la práctica profesional, trabajo en equipo y generación de código legible y portable.

3. Programa sintético

Programación estructurada y modular. Programas recursivos. Representación de datos en memoria. Paso de variables por valor y por referencia. Tipos abstractos de datos. Estructuras de datos. Estructuras contenedoras: Vectores, matrices, pilas, colas, lista, diccionarios, árboles y grafos. Implementación de estructuras de datos estáticas. Uso dinámico de memoria. Listas y árboles implementados con punteros. Análisis, eficiencia e implementación de estructuras de datos. Algoritmos de recorrido, búsqueda y ordenamiento. Nociones básicas de algoritmos sobre grafos.

4. Programa analítico

4.1 Organización del contenido

Unidad 1: Introducción

- Incorporar conceptos básicos necesarios para escribir, compilar y ejecutar un programa.
- Utilizar reglas sintácticas. Tipos de datos primitivos y referenciales.
- Definición de variables. Operadores. Estructuras de control.
- Funciones y paso de parámetros. Ámbito de variables. Sobrecarga de operadores.
- Proporcionar ejemplos y realizar ejercicios.

Unidad 2: Tipos de datos abstractos

- Incorporar el concepto de abstracción de datos y encapsulamiento de datos.
- Incorporar el concepto de tipo abstracto de dato (TDA).
- Diferenciar entre tipo de dato y tipo abstracto de dato.
- Realizar requerimientos y diseño de un TDA.
- Trabajar con TDAs

Unidad 3: Funciones recursivas, arreglos unidimensionales y bidimensionales

- Incorporar principio de recursión. Funcionamiento de recursión. Trabajar problemas y ejercicios con soluciones recursivas.

- Trabajar con arreglos unidimensionales y bidimensionales, representación en memoria, creación, recorrido, inserción y eliminación de elementos.

Unidad 4: Pilas dinámicas, definición y terminología

- Incorporar definición y terminología de pilas.
- Trabajar con el TDA pila. Operaciones básicas.
- Realizar implementación dinámica mediante listas.

Unidad 5: Colas dinámicas, definición y terminología

- Incorporar definición y terminología de colas.
- Trabajar con el TDA cola. Operaciones básicas.
- Realizar implementación dinámica mediante listas.
- Colas simples y circulares.

Unidad 6: Diccionarios

- Incorporar definición y terminología de diccionarios
- Trabajar con TDA Diccionario. Operaciones básicas.
- Implementación utilizando conjuntos y tuplas.

Unidad 7: Listas

- Incorporar definición y terminología de listas
- Trabajar con TDA lista. Operaciones básicas.
- Implementación dinámica utilizando punteros.
- Listas simples enlazadas, dobles enlazadas y circulares

Unidad 8: Árboles

- Incorporar definición y terminología de árbol.
- Árboles binarios. Árboles binarios de búsqueda. Realizar implementación utilizando punteros. Recorridos e implementaciones recursivas. Inserción, eliminación, modificación y búsqueda de elementos.
- Árboles balanceados. Árboles AVL. Árboles de M-vías.
- Trabajar con árboles y sus aplicaciones.

Unidad 9: Grafos

- Incorporar definición y terminología de grafos.
- Realizar recorridos básicos, aplicaciones y ejemplos.

4.2 Trabajos prácticos

Trabajo Práctico 1: Algoritmos de Búsqueda y Ordenamiento

En este trabajo práctico se propone la integración de los temas de la primera parte del programa, abordando los problemas de búsqueda y ordenamiento y utilizando el tipo de dato arreglo como modelo.

OBJETIVOS:

- Integrar los conceptos de recursividad y arreglos.
- Comprender los conceptos de algoritmos de búsqueda y ordenamiento.
- Implementar los algoritmos.
- Comparar los algoritmos según su complejidad temporal, para el caso promedio.

4.3 Bibliografía y recursos obligatorios:

- Roberto Tamassia, Michael H. Goldwasser & Michael T. Goodrich. (2013) "Data Structures and Algorithms in Python", Wiley (1ra Edición)
- Alfred Aho & Jeffrey Ullman. (1983) "Data Structures and Algorithms", Addison – Wesley.
- Chris Okasaki. (1998) "Purely Functional Data Structures", Cambridge University Press.
- Mark Allen Weiss. (1997) "Data Structures and Algorithms in C". Addison – Wesley (2da Edición).

5. Metodología de enseñanza

Las clases se desarrollan en una modalidad mixta (presencial – virtual) en ciclos semanales. Utilizando el soporte de la plataforma virtual del Campus de la Universidad se articulan clases teóricas virtuales asincrónicas cortas, en las cuales se verterán los conceptos básicos de cada uno de los temas en cuestión, con clases teórico - prácticas de repaso virtuales sincrónicas con ejemplos prácticos y en la que se favorece la participación de los/las estudiantes. Ambas clases virtuales quedan disponibles en el Campus virtual como material de repaso y consulta de forma permanente. Se finalizan cada uno de los temas con una clase práctica presencial en la que se resuelven situaciones problemáticas en el pizarrón y se plantean casos de uso en situaciones reales.

La mayor cantidad de tiempo se dedica a la parte práctica, con la resolución de problemas en el pizarrón de forma constructiva con participación de los/las estudiantes, permitiendo la discusión e interacción y principalmente con la realización de ejercicios por parte de los/las estudiantes en el laboratorio de computación.

Cada unidad perteneciente a los contenidos mínimos posee una guía de ejercicios asociada y es de carácter obligatorio la realización de actividades de autoevaluación semanales a través del campus virtual. Para la implementación de los ejercicios se utilizan lenguajes de programación de acuerdo al estado del arte de la tecnología, que se adecuen a los propósitos pedagógicos de la materia.

Las guías de ejercicios/autoevaluación están redactadas considerando un lenguaje inclusivo que no prioriza un género sobre otro.

6. Plan de trabajo en el campus

Se utiliza la plataforma virtual que provee la Universidad para el armado de clases teóricas virtuales, para la exposición de algunos temas, con el fin de facilitar el aprendizaje y uso de herramientas informáticas, potenciando además la autonomía, la responsabilidad y la comunicación interactiva en los estudiantes.

Los estudiantes contarán para la asignatura con una plataforma *Moodle*, a través del campus virtual de la UNA HUR.

En dicha plataforma, los estudiantes tienen disponible desde el inicio de clases: el programa, el cronograma y la bibliografía de la materia. Previo al dictado presencial de las Unidades, se habilita para cada una de éstas:

- *El material soporte de cada clase.* Consiste en las presentaciones de las teóricas y la bibliografía adicional correspondiente.
- *Material audiovisual.* Se pondrán a disposición de las/los estudiantes videos explicativos de las temáticas correspondientes, para mejorar el aprendizaje de los contenidos teóricos desarrollados en las clases. En caso de modalidad enteramente virtual se pondrán a disposición videos correspondientes a todas las clases teóricas de la materia.
- *Foros proactivos de discusión.* Las/Los docentes propondrán foros de discusión por Unidad Temática, mediante preguntas disparadoras, artículos publicados, noticias u otras herramientas que fomenten la participación activa de las/los estudiantes. Los foros podrán ser utilizados también por las/los estudiantes para la realización de consultas e intercambios con las/los docentes y/o entre ellas/ellos mismas/mismos.
- *Ejercicios y actividades de autoevaluación.* Se ponen a disposición de las/los estudiantes ejercicios prácticos y cuestionarios en formato *multiple choice* correspondientes a las diferentes clases teóricas. Esto permitirá a las/los estudiantes ejercitarse y evaluar el grado de aprendizaje adquirido. Se podrá realizar en la cantidad de intentos que cada estudiante desee, dado que el resultado obtenido no será tenido en cuenta para su consideración por parte del cuerpo docente de la asignatura.

7. Evaluación y régimen de aprobación

7.1 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

La/El estudiante deberá poseer una asistencia no inferior al 75% en las clases presenciales.

En cuanto a las cursadas virtuales se requerirá que la/el estudiante ingrese al aula virtual como mínimo una vez por semana.

7.2 Aprobación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: Tal como lo establece el Art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no 7 inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El promedio estricto resultante deberá ser una nota igual o superior a siete(7) sin mediar ningún redondeo.

Evaluación integradora: Tal como lo establece el Art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos/as estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos. La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia. La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.

7.3 Criterios de calificación

La calificación de las/los estudiantes se compone de las notas de los parciales (con sus respectivos recuperatorios), haciendo foco en las resoluciones conceptuales y otorgando menor peso a errores de codificación, intentando no evaluar los conocimientos de un lenguaje de programación en particular. Se toma en cuenta como nota de concepto el trabajo en clase y las entregas obligatorias de ejercicios de autoevaluación semanal.

Trabajo Práctico e Informe. Se realiza un Trabajo Práctico sobre la temática “Búsqueda y Ordenamiento”, este trabajo se realiza y entrega en grupos de hasta 3 estudiantes. Las/Los estudiantes deben expresar los resultados obtenidos en un informe del que se evaluará la

comprensión de lo observado y claridad expositiva. El informe obligatorio se califica de manera numérica y la nota se toma como nota de concepto para tener en cuenta al momento de cerrar la nota de la cursada.

8. Cronograma

8.1 Estructura de clases y ciclos semanales

Cada tema correspondiente al contenido del programa analítico se desarrolla en una o dos semanas según la duración y complejidad del mismo. Cada ciclo semanal se compone de 3 instancias en una modalidad mixta virtual - presencial con la siguiente estructura:

- Clase teórica virtual asincrónica
- Clase teórico-práctica virtual sincrónica
- Clase práctica presencial

8.2 Cronograma semanal

Semana	Temario
1	Unidad 1: Presentación de los contenidos y temas de la materia y estructura de las clases. Lenguajes de programación. Escritura, compilación y ejecución. Tipos de datos primitivos. Lenguaje de programación Python. Operadores básicos. Comunicación con usuario. Primeros ejemplos prácticos y ejercicios en PC.
2	Unidad 1: Lenguaje de programación Python. Estructuras de control. Definición de funciones y paso de parámetros. Ejemplos prácticos y ejercicios en PC.
3	Unidad 2: Concepto de tipo de dato abstracto (TDA). Requerimientos y diseños de TDA. Diferentes tipos de TDA. Ejemplos prácticos y ejercicios en PC.
4	Unidad 3: Programación modular, Programación estructurada. Resoluciones iterativas. Introducción a la Recursividad. Implementaciones recursivas. Ejemplos prácticos y ejercicios en PC.
5	Unidad 3: Cadenas. Arreglos unidimensionales. Arreglos multidimensionales. Operaciones básicas. Asignación de memoria. Ejemplos prácticos y ejercicios en PC.
6	Trabajo Práctico Búsqueda y Ordenamiento
7	Unidad 4: Pilas y Colas dinámicas. Definición y operaciones básicas. Implementación con listas. Ejemplos prácticos y ejercicios en PC.
8	Unidad 5: Diccionarios y conjuntos. Definición y operaciones básicas. Implementación. Ejemplos prácticos y ejercicios en PC.
9	Primer Parcial
10	Unidad 6: Estructuras dinámicas. Listas. Definición y terminología. Operaciones básicas, agregar,

	borrar, intercambiar elementos, etc. Ejemplos prácticos y ejercicios en PC.
11	Unidad 6: Estructuras dinámicas. Listas. Definición y terminología. Operaciones básicas, agregar, borrar, intercambiar elementos, etc. Ejemplos prácticos y ejercicios en PC.
12	Unidad 6: Listas. Implementación recursiva.
13	Unidad 7: Árboles. Definición y terminología. Árboles binarios. Árboles binarios de búsqueda. Inserción en árboles binarios. Definiciones de peso, altura, etc. Ejemplos prácticos y ejercicios en PC.
14	Unidad 7: Árboles binarios de búsqueda. Borrado en árboles binarios, Ejemplos prácticos y ejercicios en PC.
15	Segundo Parcial
16	Recuperatorios